

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-278952

(43)Date of publication of application : 27.09.2002

(51)Int.Cl.

G06F 17/21
G06F 3/14

(21)Application number : 2001-074178

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 15.03.2001

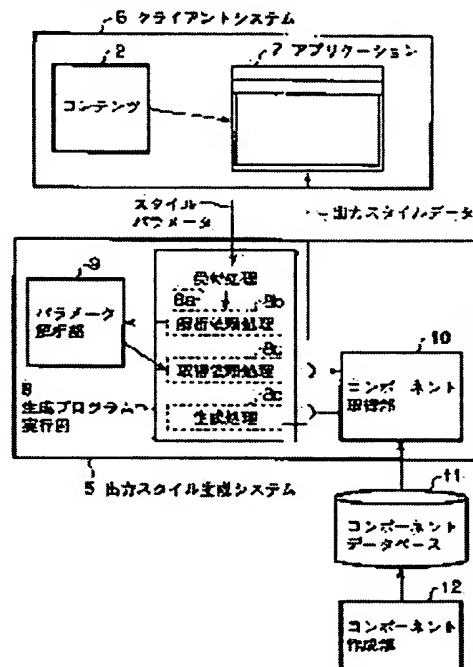
(72)Inventor : YANO REI
SHIMAMURA CHIZUKO
HARASHIMA HIDEJI

(54) OUTPUT STYLE GENERATING PROGRAM AND METHOD AND DEVICE FOR GENERATING OUTPUT STYLE

(57)Abstract:

PROBLEM TO BE SOLVED: To easily and speedily obtain defining the output style of a content.

SOLUTION: A computer for generating the output style of the content 2 where the content of data is constituted based on at least one element functions as a reception means 8a receiving a style parameter designating the output styles for the respective elements constituting the content 2, a component acquisition means 10 acquiring a component that the style parameter received by the reception means 8a designates from the component group 11 which is previously set as the output style and a generating means 8d generating output style data corresponding to the style parameter based on the component acquired by the component acquisition means 10, by an output style generating program.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

This Page Blank (uspto)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-278952
(P2002-278952A)

(43) 公開日 平成14年9月27日 (2002.9.27)

(51) Int.Cl. ⁷	識別記号	F I	テームコード [*] (参考)
G 0 6 F 17/21	5 3 0	G 0 6 F 17/21	5 3 0 E 5 B 0 0 9
			5 3 0 T 5 B 0 6 9
3/14	3 1 0	3/14	3 1 0 C

審査請求 未請求 請求項の数 6 O L (全 15 頁)

(21) 出願番号 特願2001-74178(P2001-74178)

(22) 出願日 平成13年3月15日 (2001.3.15)

(71) 出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72) 発明者 矢野 令

東京都府中市東芝町1番地 株式会社東芝
府中事業所内

(72) 発明者 島村 千鶴子

東京都府中市東芝町1番地 株式会社東芝
府中事業所内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

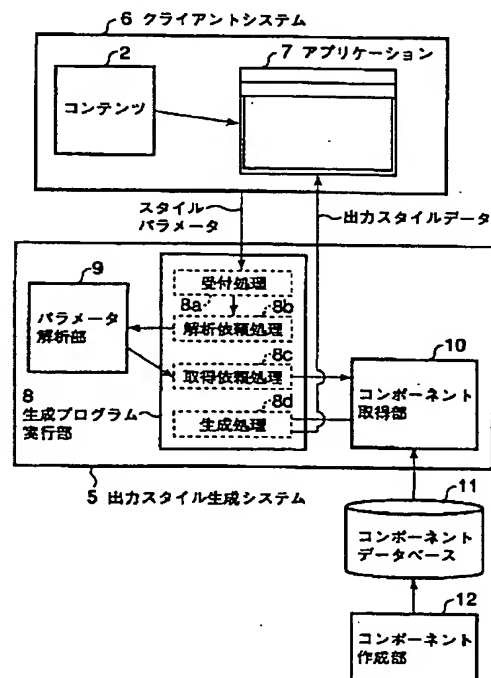
最終頁に続く

(54) 【発明の名称】 出力スタイル生成プログラム及び出力スタイル生成方法並びに出力スタイル生成装置

(57) 【要約】

【課題】 コンテンツの出力スタイルを定義するデータを容易かつ迅速に取得する。

【解決手段】 少なくとも一つの要素に基づいてデータの内容が構成されるコンテンツ2の出力スタイルを生成するためにコンピュータを、コンテンツ2を構成する要素毎の出力スタイルを指定するスタイルパラメータを受け付ける受付手段8a、予め出力スタイルとして設定されているコンポーネント群11から受付手段8aに受け付けられたスタイルパラメータの指定するコンポーネントを取得するコンポーネント取得手段10、コンポーネント取得手段10によって取得されたコンポーネントに基づいてスタイルパラメータに対応する出力スタイルデータを生成する生成手段8d、として機能させるための出力スタイル生成プログラム。



【特許請求の範囲】

【請求項1】 少なくとも一つの要素に基づいてデータの内容が構成されるコンテンツの出力スタイルを生成するためにコンピュータを、

あるコンテンツを構成する要素毎の出力スタイルを指定するスタイルパラメータを受け付ける受付手段、

予め出力スタイルとして設定されているコンポーネント群の中から前記受付手段に受け付けられたスタイルパラメータの指定するコンポーネントを取得するコンポーネント取得手段、

前記コンポーネント取得手段によって取得されたコンポーネントに基づいて前記スタイルパラメータに対応する出力スタイルデータを生成する生成手段、として機能させるための出力スタイル生成プログラム。

【請求項2】 請求項1記載の出力スタイル生成プログラムにおいて、

前記生成手段によって生成された出力スタイルデータを利用した出力を実行し、この出力内容の更新部分のスタイルパラメータを受け付けた場合に、この更新部分のスタイルパラメータと前記受付手段と前記コンポーネント取得手段と前記生成手段とを利用して更新後の出力スタイルデータを求め、更新後の出力を実行するフィードバック手段としてコンピュータを機能させることを特徴とする出力スタイル生成プログラム。

【請求項3】 請求項1又は請求項2記載の出力スタイル生成プログラムにおいて、

前記受付手段によって受け付けられたスタイルパラメータを解析し、所定の形式に変換するパラメータ解析手段としてコンピュータを機能させ、

前記コンポーネント取得手段は、前記パラメータ解析手段による解析後のスタイルパラメータに基づいてコンポーネントを取得することを特徴とする出力スタイル生成プログラム。

【請求項4】 請求項1乃至請求項3のいずれか1項記載の出力スタイル生成プログラムにおいて、前記コンポーネント群は、コンポーネントデータベースに登録されており、

前記生成手段は、生成した出力スタイルデータに固有のパラメータを対応付けてコンポーネントとして前記コンポーネントデータベースに登録することを特徴とする出力スタイル生成プログラム。

【請求項5】 少なくとも一つの要素に基づいてデータの内容が構成されるコンテンツの出力スタイルを生成するコンピュータシステムによる出力スタイル生成方法であって、

あるコンテンツを構成する要素毎の出力スタイルを指定するスタイルパラメータを受け付ける工程、

予め出力スタイルとして設定されているコンポーネント群の中から受け付けられたスタイルパラメータの指定するコンポーネントを取得する工程、

取得されたコンポーネントに基づいて前記スタイルパラメータに対応する出力スタイルデータを生成する工程、を含むことを特徴とする出力スタイル生成方法。

【請求項6】 あるコンテンツを構成する要素毎の出力スタイルを指定するスタイルパラメータを受け付ける受付手段と、

予め出力スタイルとして設定されているコンポーネント群の中から前記受付手段に受け付けられたスタイルパラメータの指定するコンポーネントを取得するコンポーネント取得手段と、

前記コンポーネント取得手段によって取得されたコンポーネントに基づいて前記スタイルパラメータに対応する出力スタイルデータを生成する生成手段とを具備したことを特徴とする出力スタイル生成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データを出力する場合のスタイルを定義する出力スタイルデータを生成する出力スタイルの生成プログラム、出力スタイル生成方法、出力スタイル生成装置に関する。

【0002】

【従来の技術】 計算機システムで扱われるデータの中には、具体的なデータの内容とそのデータの内容を表示・印刷・操作する場合のスタイルとが混合して記述されているデータがある。

【0003】 計算機システムは、データの内容とスタイルとが混合して記述されているデータを表示・印刷・操作する場合、このデータで記述されている内容をこのデータで記述されているスタイルにしたがって表示・印刷・操作する。

【0004】 一方、構造化データや非構造化データ（以下、あわせてコンテンツという）と、コンテンツに対応する出力スタイルデータとを、計算機システムが分離して扱う場合もある。構造化データの例としてはXML（eXtensible Markup Language）により記述されたデータがある。

【0005】 このように、コンテンツと出力スタイルデータとを分離して扱うことで、あるコンテンツに対して複数の出力スタイルデータを用意しておき、同じ内容のコンテンツであっても複数のスタイルで表示・印刷・操作することが可能である。

【0006】 図19は、計算機システムにおけるコンテンツと出力スタイルデータの取り扱い状態を示すブロック図である。ブラウザ1は、コンテンツ2の内容を出力スタイルデータ3で指定されているスタイルで表示する。

【0007】

【発明が解決しようとする課題】 しかしながら、出力スタイルデータは所定のルールにしたがって所定の言語で記述する必要がある。このため、出力スタイルデータを

記述するには専門的な知識が必要であり、出力スタイルデータの生成には手間がかかる。また、コンテンツの内容の変更やスタイルの変更に柔軟に対処することが困難である。さらに、あるコンテンツに対応した出力スタイルデータを他のコンテンツに対応した出力スタイルデータに再利用することが困難である。

【0008】そのため、従来においては、図20に示すように、あるコンテンツ2に対応する数種類の出力スタイルデータ3a~3cを静的に用意し、例えばユーザはそのうちのいずれかを選択して利用していた。このように、コンテンツに対応する出力スタイルデータが予め限定されてしまうと、柔軟にスタイルを変更することが困難となる。

【0009】以上の問題を解決するための技術が特開平11-143874号に開示されている。

【0010】図21は、特開平11-143874号に記載されている構造化文書の体裁定義支援装置のシステム構成図である。

【0011】この構造化文書の体裁定義支援装置4は、構造化情報の構成要素に対応して体裁情報の雛形の一覧をユーザに提示し、ユーザに選択された雛形の内容を取得して構成要素毎の体裁定義情報を生成し、この構成要素毎の体裁定義情報の集まりをスタイルシート（出力スタイルデータ）とする。

【0012】しかしながら、この構造化文書の体裁定義支援装置4では、構造化情報の構成要素のすべてについて体裁情報の一覧の提示が行われ、ユーザは構成要素のすべてについて体裁情報の選択を行う必要がある。

【0013】したがって、ユーザは構成要素の数だけ繰り返し入力操作を行う必要があり、スタイルシートの生成に時間がかかり、迅速にスタイルシートを生成すること、及び多数のスタイルシートを生成することが困難である。また、ユーザの操作労力が大きくなる。そして、構成要素の数が増加するほどスタイルシートの生成時間やユーザの労力がかかる。

【0014】ゆえに、構造化文書の体裁定義支援装置4を利用しても、個々のユーザの好みに応じて、あるいは状況に応じて動的にスタイルを生成するという効果を得ることは困難である。

【0015】また、この構造化文書の体裁定義支援装置4を利用した場合、ユーザの選択操作が必要であるためプログラム上で自動的に出力スタイルデータを作成することができない。

【0016】さらに、この構造化文書の体裁定義支援装置4ではスタイルシートの生成時に構造化情報（コンテンツ）を入力する必要があるため、スタイルシートの生成時に構造化情報が取得されている必要がある。しかしながら、構造化情報を取得する前に、予めスタイルシートを準備しておきたい場合もある。

【0017】本発明は、以上のような実情に鑑みてな

れたもので、コンテンツに対応する出力スタイルデータの動的生成を可能とする出力スタイル生成プログラム、出力スタイル生成方法、出力スタイル生成装置を提供することを目的とする。

【0018】

【課題を解決するための手段】以下、本発明を実現するにあたって講じた具体的手段について説明する。

【0019】本発明は、少なくとも一つの要素に基づいてデータの内容が構成されるコンテンツの出力スタイルを生成するためにコンピュータを、あるコンテンツを構成する要素毎の出力スタイルを指定するスタイルパラメータを受け付ける受付手段、予め出力スタイルとして設定されているコンポーネント群の中から受付手段に受け付けられたスタイルパラメータの指定するコンポーネントを取得するコンポーネント取得手段、コンポーネント取得手段によって取得されたコンポーネントに基づいてスタイルパラメータに対応する出力スタイルデータを生成する生成手段、として機能させるための出力スタイル生成プログラムである。

【0020】本発明においては、入力されたスタイルパラメータに基づいてコンポーネント化された出力スタイルが組み合わせられ、出力スタイルデータが生成される。

【0021】したがって、スタイルパラメータを1回入力するのみで容易かつ迅速に出力スタイルを自動生成できる。これにより、出力スタイルデータの生成に必要な入力回数を制限することができる。

【0022】また、迅速かつ容易に出力スタイルデータを生成できるため、状況に応じて、あるいはユーザの要求する時点で、スタイルパラメータを入力し、出力スタイルデータを生成することができる。

【0023】これにより、例えばユーザ毎の出力スタイルデータ、あるいは状況に応じた出力スタイルデータを動的生成し、同じ内容のコンテンツであってもユーザ毎のスタイル、あるいは状況に応じたスタイルで表示・印刷・操作することができる。すなわち、出力スタイルの自由度を拡大できる。

【0024】また、出力スタイルデータの生成時に、コンテンツ自体を必要としないため、コンテンツ取得前であっても予め出力スタイルデータを生成しておくことができる。

【0025】上記発明は、各種変形可能である。

【0026】第1の変形例として、生成手段によって生成された出力スタイルデータを利用して特定のコンポーネント又は仮想的なデータの出力を実行し、この出力内容の更新部分のスタイルパラメータを受け付けた場合に、この更新部分のスタイルパラメータと受付手段とコンポーネント取得手段と生成手段とを利用して更新後の出力スタイルデータを求め、特定のコンポーネント又は仮想的なデータの更新後の出力を実行するフィードバック手段としてコンピュータを機能させるとしてもよい。

【0027】なお、仮想的なデータとは、出力スタイルを確定するために利用されるデータである。

【0028】これにより、ユーザは入力済みのスタイルパラメータのうち更新部分のスタイルパラメータのみを入力することで出力スタイルデータを繰り返し更新することができ、ユーザの好みに適した出力スタイルデータを容易かつ迅速に生成できる。

【0029】第2の変形例として、受付手段によって受け付けられたスタイルパラメータを解析し、所定の形式に変換するパラメータ解析手段としてコンピュータを機能させ、コンポーネント取得手段は、パラメータ解析手段による解析後のスタイルパラメータに基づいてコンポーネントを取得するとしてもよい。

【0030】なお、所定の形式とは、生成手段やコンポーネント取得手段で取り扱い可能な形式をいう。

【0031】これにより、受け付けられたスタイルパラメータの形式を自動的に修正できる。

【0032】第3の変形例として、コンポーネント群は、コンポーネントデータベースに登録されており、生成手段は、生成した出力スタイルデータに固有のパラメータを対応付けてコンポーネントとしてコンポーネントデータベースに登録するとしてもよい。

【0033】これにより、柔軟に出力スタイルデータを再利用できる。

【0034】このようなプログラム又はこのプログラムを記録した記録媒体を用いることによって上述した手段を備えていないサーバ、計算機、計算機システムに対しても、簡単に本発明の手段を付加することができ、本発明の作用効果を得ることができる。

【0035】なお、上記の受付手段、コンポーネント取得手段、生成手段、フィードバック手段、パラメータ解析手段を具備した出力スタイル生成装置を発明の対象としてもよい。

【0036】また、上記の受付手段、コンポーネント取得手段、生成手段、フィードバック手段、パラメータ解析手段の実行内容を各工程とするコンピュータシステムによる出力スタイル生成方法を発明の対象としてもよい。

【0037】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態について説明する。

【0038】（第1の実施の形態）本実施の形態においては、スタイルパラメータに基づいてコンポーネント化された出力スタイルを組み合わせて出力スタイルデータを生成する出力スタイル生成システムについて説明する。

【0039】ここで、スタイルパラメータとは、コンテンツを構成する要素毎に、その出力スタイルを指定したパラメータである。

【0040】また、出力スタイルとしては、例えば表示

色、表示する場合の配置位置、表示する字の大きさ、文字飾り、文字の形、テーブル表示、リスト表示、グラフ表示などがあり、さらに通信のフォーマット、音声出力、プリンタ印刷、FAX送信、画面表示、点字による出力なども含む。

【0041】図1は、本実施の形態に係る出力スタイル生成システムの構成を例示するブロック図である。

【0042】出力スタイル生成システム5を利用するクライアントシステム6は、例えばXMLなどで記述されたコンテンツ2を保持している。また、クライアントシステム6は、出力スタイルデータの内容にしたがってコンテンツの表示・印刷・操作を行う処理を実行するアプリケーション7を備えている。アプリケーション7の一例には、ブラウザがある。

【0043】出力スタイル生成システム5は、クライアントシステム6からスタイルパラメータを入力し、このスタイルパラメータに応じた出力スタイルデータを生成してクライアントシステム6に出力する。

【0044】この出力スタイル生成システム5は、主に、各種処理8a～8dを実行する生成プログラム実行部8、パラメータ解析部9、コンポーネント取得部10とから構成されており、必要に応じてコンポーネントデータベース11をアクセスする。

【0045】コンポーネントデータベース11は、出力スタイルデータを生成するために必要な雛形であるコンポーネントを多数格納している。なお、コンポーネントとは、例えばファイルやデータベースに格納されているデータや直列化されたプログラムオブジェクトなどである。

【0046】コンポーネント作成部12は、例えばGUIなどによりコンポーネントの作成を支援し、作成されたコンポーネントをコンポーネントデータベース11に格納する。

【0047】パラメータ解析部9は、生成プログラム実行部8からの命令に基づいてスタイルパラメータを解析し、生成プログラム実行部8で取り扱い可能な形式に変換し、生成プログラム実行部8に返す。このパラメータ解析部9は、ある特定のパラメータ形式のみではなく、必要に応じて様々なスタイルパラメータを解析し、所定の形式に変換可能である。

【0048】生成プログラム実行部8は、プログラム実行部であり、クライアントシステム6からのスタイルパラメータの受付処理8aを実行し、パラメータ解析部9にスタイルパラメータの解析を依頼する解析依頼処理8bを実行する。

【0049】また、生成プログラム実行部8は、パラメータ解析部9で解析されたスタイルパラメータに基づいてコンポーネント取得部10にコンポーネントの取得を依頼する依頼処理8cを実行し、この結果に基づいて出力スタイルデータを自動生成する生成処理8dを実行す

る。出力スタイルデータの自動生成のためのロジックは多数存在するため、生成プログラム実行部も複数存在しえる。

【0050】コンポーネント取得部10は、生成プログラム実行部8からの命令に基づいてコンポーネントデータベース11から該当するコンポーネントを取得し、生成プログラム実行部8に返す。また、コンポーネント取得部10は、コンポーネントデータベース11から読み出したコンポーネントに基づいてプログラムを実行し、その結果を生成プログラム実行部8に返す。

【0051】以下に、パラメータ解析部9、生成プログラム実行部8、コンポーネント取得部10の処理内容を具体的に説明する。

【0052】図2は、パラメータ解析部9の処理内容を例示する図である。

【0053】例えば、パラメータ解析部9は、「param1=A,param2=B,param3=C」という形式のスタイルパラメータparamsを引数として受け付けると、このスタイルパラメータparamsをカンマで区切って「param1=A」「param2=B」「param3=C」とし、ハッシュテーブルに格納する。

【0054】そして、パラメータ解析部9は、ハッシュテーブルから順次「param1=A」「param2=B」「param3=C」を取り出し、「=」の左辺をキー、右辺を値として(キー、値)という形式に変換し、ハッシュテーブルに格納する。これにより、「((param1,A)(param2,B)(param3,C))」という解析済みのスタイルパラメータがハッシュテーブルに格納される。

【0055】パラメータ解析部9は、この解析済みのスタイルパラメータを格納したハッシュテーブルを返す。

【0056】図3は、生成プログラム実行部8の処理内容を例示する図である。

【0057】例えば、生成プログラム実行部8は、「param1=A,param2=B,param3=C」という形式のスタイルパラメータparamsを引数として受け付けると(受付処理8a)、スタイルパラメータparamsをパラメータ解析部9に渡して解析を依頼し(解析依頼処理8b)、解析済みのスタイルパラメータ「((param1,A)(param2,B)(param3,C))」を取得する。

【0058】そして、生成プログラム実行部8は、「param1」について値「A」、「param2」について「B」、「param3」について「C」に対応するコンポーネントの取得をコンポーネント取得部に依頼し(取得依頼処理8c)、取得したコンポーネントをバッファに格納し、出力スタイルデータとして返す(生成処理8d)。

【0059】図4は、コンポーネント取得部10の処理内容を例示する図である。

【0060】例えば、コンポーネント取得部10は、命令にしたがってコンポーネントデータベース11を参照し、複数のコンポーネントを取り出し、組み合わせて返す処理10aを実行する。

【0061】また、コンポーネント取得部10は、命令にしたがってコンポーネントデータベース11を参照し、受け付けた引数をパラメータとしてコンポーネントを取り出し、コンポーネントに含めて返す処理10bを実行する。

【0062】図5は、コンポーネント作成部12のGUI画面の例を示す図である。

【0063】この画面12aにより要素の形式、サイズ、文字飾り、位置を指定し、コンポーネントの名前を付けると、指定された出力スタイルを実現するためのコンポーネントがコンポーネントデータベース11に格納される。

【0064】図6は、コンポーネントデータベース11に登録されているコンポーネントの例である。

【0065】本実施の形態におけるコンポーネントにはワイルドカード(この例ではアスタリスク)が利用されている。これにより、コンポーネントを特定のコンテンツに依存しないスタイルとすることが可能となる。

【0066】コンポーネント11a、11bは、パラメータを含める必要のない出力スタイルを定義するデータである。コンポーネント11c、11dは、パラメータを含める必要があり{}の中にパラメータを含める出力スタイルを定義するデータである。

【0067】上記出力スタイル生成システム5の実現する工程を以下に説明する。

【0068】図7は、出力スタイル生成システム5の実行する処理を示すフロー図である。

【0069】出力スタイル生成システム5は、クライアントシステム6からスタイルパラメータを受け付けると(ST1)、生成プログラム実行部8を起動し(ST2)、パラメータ解析部9によりスタイルパラメータを解析する(ST3)。すると、生成プログラム実行部8は、解析後のスタイルパラメータに対する処理を開始する(ST4)。

【0070】生成プログラム実行部8は、解析後のスタイルパラメータに含まれる全てのパラメータに対して処理を実行したか否かを判定し(ST5)、未処理のパラメータがあればそのパラメータの値が正当か否かをチェックする(ST6)。

【0071】未処理で値の正当なパラメータがあれば生成プログラム実行部8は、コンポーネント取得部10に取得を依頼し、その際にパラメータが必要であれば生成プログラム実行部8からコンポーネント取得部10にパラメータが渡される。

【0072】コンポーネント取得部10は、生成プログラム実行部8からの要求に応じてコンポーネントデータベース11からコンポーネントを取得し(ST7)、生成プログラム実行部8に返す。

【0073】生成プログラム実行部8は、コンポーネント取得部10によって取得されたコンポーネントをバッフ

ァに格納して組み合わせ（ST8）、出力スタイルデータとしてクライアントシステム6に返す（ST9）。

【0074】以上説明した本実施の形態に係る出力スタイル生成システム5においては、スタイルパラメータの入力という必要最小限のイベントにより容易かつ迅速に出力スタイルデータが自動生成できる。また、出力スタイルデータの生成に必要な入力回数を制限できる。

【0075】したがって、同一内容のコンテンツについて、ユーザ毎にあるいは状況に応じて自由に出力スタイルを変更できる。例えば、同一内容のコンテンツであっても通常のスタイル、予め指定された部分だけを表示するスタイル、重要な部分を音声で読み上げ他の部分は画面に表示するスタイル、所定のユーザに対する点字出力などのように、様々な出力方法を自由に提供できる。

【0076】また、出力スタイルデータの生成時にコンテンツ自体は必要ないため、例えばコンテンツを構成する要素は分かっているがその具体的な内容が未知のコンテンツについて予め出力スタイルデータを生成しておくこともできる。

【0077】本実施の形態に係る出力スタイル生成システム5は、文書、表、写真など様々な形式で表現されているデータを検索結果としてまとめて表示する場合（例えば電子カタログ、検索エンジン、ブラウザなどの表示用GUI、XMLデータベースの表示用GUI）に利用する出力スタイルデータの生成に特に有効である。

【0078】また、本実施の形態に係る出力スタイル生成システム5は、表示の仕方は定まっているが画像などの様々なデータをそれぞれ異なる方式で表示しなければならない場合（例えば電子カタログ、電子商取引における取引データの一覧表示、ブラウザなどの表示用GUI、XMLデータベースの表示用GUI）に利用する出力スタイルデータの生成に特に有効である。

【0079】なお、本実施の形態においては、クライアントシステム6がコンテンツを保持しているが、コンテンツは他の装置が保持していてもよい。例えば、出力スタイル生成システム5が出力スタイルデータとともにコンテンツを保持し、任意の提供先にそのコンテンツと出力スタイルデータとを提供するとしてもよい。

【0080】また、クライアントシステム6ではなくユーザが出力スタイル生成システム5にスタイルパラメータを入力するとしてもよい。

【0081】（第2の実施の形態）本実施の形態においては、上記第1の実施の形態に係る出力スタイル生成システム5の実行する処理について具体例を用いて詳細に説明する。

【0082】ここでは、図8に示すような商品リストの構造化データを出力対象のコンテンツ13とし、スタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」に基づいて出力スタイルデータを生成する場合について説明する。

【0083】図9は、スタイルパラメータを受け付けた出力スタイル生成システム5の具体的な処理内容を示すフロー図である。

【0084】出力スタイル生成システム5は、クライアントシステム6からスタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」を受け付けると（ST10）、該当する生成プログラム実行部8を起動し（ST11）、この生成プログラム実行部8にスタイルパラメータを渡す。すると、スタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」について図10に示す生成プログラム実行部8の処理が開始される。

【0085】生成プログラム実行部8は、パラメータ解析部9にスタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」を渡し、解析を依頼する。すると、スタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」について図11に示すパラメータ解析処理部9の処理が開始される。

【0086】パラメータ解析部9はスタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」を解析し（ST13）、生成プログラム実行部8で扱える形式のスタイルパラメータ「（（見出し、下線付き見出し）（表現形式、テーブル））」を生成プログラム実行部8に返す。すると、解析後のスタイルパラメータ「（（見出し、下線付き見出し）（表現形式、テーブル））」に対する図10に示す生成プログラム実行部8の処理が開始される（ST14）。

【0087】生成プログラム実行部8は、まずパラメータ「見出し」の値をチェックし（ST15）、この値「下線付き見出し」に該当するコンポーネントの取得をコンポーネント取得部10に依頼する。すると、図12に示すコンポーネント取得部10の該当する処理10cが実行される。

【0088】コンポーネント取得部10は、コンポーネントデータベース11からコンポーネントを取得して生成プログラム実行部8に返し（ST16a）、生成プログラム実行部8はコンポーネントをバッファに追加する（ST17a）。

【0089】次に、生成プログラム実行部8は、パラメータ「表示形式」の値をチェックし（ST18）、この値「テーブル」に該当するコンポーネントの取得をコンポーネント取得部10に依頼する。すると、図12に示すコンポーネント取得部10の該当する処理10eが実行される。

【0090】コンポーネント取得部10は、コンポーネントデータベース11からコンポーネントを取得して生成プログラム実行部8に返し（ST19a）、生成プログラム実行部8はコンポーネントをバッファに追加する（ST20a）。

【0091】そして、生成プログラム実行部8は、取得したコンポーネントを結合して出力スタイルデータを生

成し(ST21)、図10に示すようにクライアントシステム6に返す(ST22)。

【0092】図13は、上記処理で利用されるコンポーネントを例示する図である。見出しを下線付きで表示するためのコンポーネント11e、見出しを斜体で表示するためのコンポーネント11f、テーブル表示のためのコンポーネント11g、リスト表示のためのコンポーネント11hがある。

【0093】図14は、スタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」に基づいて上記処理により生成された出力スタイルデータを用いて上記図8のコンテンツ13を表示した画面図である。画面14aでは、コンテンツ13の内容のうち見出しが下線付きで表示され、商品名と価格とがテーブルで表示されている。

【0094】一方、図15は、スタイルパラメータ「見出し=斜体見出し、表示形式=リスト」に基づいて生成された出力スタイルデータを用いて上記図8のコンテンツ13を表示した画面図である。画面14bでは、コンテンツ13の内容のうち見出しが斜体で表示され、商品名と価格とがリストで表示されている。

【0095】このように、出力スタイル生成システム5を利用すれば、スタイルパラメータを入力するのみで容易かつ迅速に自己の望む出力スタイルデータを得ることができる。

【0096】(第3の実施の形態) 本実施の形態においては、生成された出力スタイルデータをさらにコンポーネントとしてコンポーネントデータベースに登録し、再利用可能とする出力システム生成システムについて説明する。

【0097】図16は、本実施の形態に係る出力スタイル生成システムの構成を例示するブロック図である。この図16において図1と同一の部分には同一の符号を付してその説明を省略する。

【0098】この出力スタイル生成システム15の生成プログラム実行部16は、取得したコンポーネントを組み立てて出力スタイルデータを生成し、この生成した出力スタイルデータに対して固有のパラメータ(例えば名称)を発行し、この固有のパラメータと生成された出力スタイルデータとを関連付けてコンポーネントデータベース11に登録する生成処理16dを実行する特徴を持つ。なお、他の処理については、上記図1の生成プログラム実行部8と同じである。

【0099】これにより、先に生成された出力スタイルデータを容易に再利用できる。

【0100】(第4の実施の形態) 本実施の形態においては、生成された出力スタイルデータによる出力結果を参考にして出力スタイルデータを更新するフィードバック部を備えた出力スタイル生成システムについて説明する。

【0101】図17は、本実施の形態に係る出力スタイル生成システムの構成を例示するブロック図である。この図17において図1と同一の部分には同一の符号を付してその説明を省略する。

【0102】この出力スタイル生成システム17に備えられているフィードバック部18は、ユーザから受け付けたスタイルパラメータに対応する出力スタイルデータの生成を生成プログラム実行部8に依頼し、その結果である出力スタイルデータを受け付ける。

【0103】すると、フィードバック部18は、最新の出力スタイルデータを利用した出力状態をユーザに提供する。

【0104】なお、フィードバック部18は、特定のコンテンツを保持する場合には、最新の出力スタイルデータを利用してその特定のコンテンツの出力状態をユーザに提供する。

【0105】一方、フィードバック部18は、特定のコンテンツを保持しない場合には、最新の出力スタイルデータを利用して仮想的なコンテンツの出力状態をユーザに提供する。

【0106】ユーザは、この出力状態を観察し、更新を望む部分についてのスタイルパラメータをフィードバック部18に入力する。

【0107】フィードバック部18は、この更新部分のスタイルパラメータに対応する出力スタイルデータの生成を生成プログラム実行部8に依頼し、その結果更新部分の出力スタイルデータを受け付ける。

【0108】そして、フィードバック部18は、先に受け付けた出力スタイルデータと更新部分の出力スタイルデータとを合成した出力スタイルデータを生成し、合成後の出力スタイルデータに基づく出力状態をユーザに提供する。

【0109】以上の操作をユーザは繰り返し実行し、自己の満足する出力状態が得られた場合に、その出力状態を実現している出力スタイルデータを出力する。

【0110】これにより、出力スタイルデータを更新したい場合に、更新部分のスタイルパラメータを入力するのみでよく、柔軟に出力スタイルデータを再利用でき、出力スタイルデータを変更できる。

【0111】例えば、ユーザはスタイル適用後の結果を観察し「見出しのサイズを1ポイント小さくする」、「テーブルの色を変更する」などスタイルの各種更新を容易に行うことができ、また更新後の結果を観察することができる。

【0112】そして、更新後の出力スタイルデータを出力し、コンテンツの出力に実用することができる。

【0113】なお、本実施の形態においては、フィードバック部18は、先の出力スタイルデータと更新部分の出力スタイルデータとを合成して更新後の出力スタイルデータを生成する手法を採用している。しかしながら、

他の手法として、先のスタイルパラメータと更新を望む部分のスタイルパラメータとから更新後のスタイルパラメータを生成し、この更新後のスタイルパラメータを生成プログラム実行部8に提供して更新後の出力スタイルデータを取得してもよい。

【0114】また、上記のフィードバック部18は、上記第3の実施の形態に係る出力スタイル生成システム15についても同様に適用可能である。

【0115】(第5の実施の形態) 本実施の形態においては、インターネットを利用してコンポーネントを送受信するネットワークシステムについて説明する。

【0116】図18は、本実施の形態に係る出力スタイル生成システム5の利用状況を例示するブロック図である。

【0117】コンポーネントデータベース11は、インターネット20上で共同利用される。

【0118】会社19Aと会社19Bと会社19Cとはコンテンツを共有するが、それぞれが異なる出力スタイルを採用している。

【0119】この場合、会社19A～19Cは、自己の採用している出力スタイルにしたがったスタイルパラメータを、インターネット20を介して出力スタイル生成システム5に入力すれば、自己の採用している出力スタイルにしたがった出力スタイルデータを取得できる。この場合、出力スタイル生成システム5の運用者は、出力スタイルデータの提供先に対価を請求してもよい。

【0120】これにより、他社からコンテンツを提供された会社、あるいは他社からコンテンツを提供される前の会社であっても、迅速かつ容易に出力スタイルデータを取得できる。

【0121】また、各会社19A～19Cで採用されている通信フォーマットを予めコンテンツとしてコンテンツデータベース11に登録しておけば、ある会社から他の会社にコンテンツを送信する場合に、この送信先の会社の通信フォーマットによりコンテンツを送信できる。

【0122】例えば、会社19Cは、会社19Aには会社19A用の通信フォーマットでコンテンツを送信し、会社19Bには会社19B用の通信フォーマットでコンテンツを送信するとする。この場合、会社19Cは、出力スタイル生成システム5に対してスタイルパラメータ「会社19A」、「会社19B」を指定するのみで自動的に各会社19A、19Bに対応した通信フォーマットを取得し、コンテンツを送信できる。

【0123】なお、本実施の形態においては、上記第1の実施の形態に係る出力スタイル生成システム5を利用する場合について説明しているが、上記第3、4の実施の形態に係る出力スタイル生成システム15、17を同様に利用することも可能である。

【0124】また、上記各実施の形態で説明した出力スタイル生成システム1、15、17においては、同様の

作用・機能を実現可能であれば各構成要素の配置を変更させてもよく、また各構成要素を自由に組み合わせてもよい。

【0125】また、上記各実施の形態で説明した出力スタイル生成システム1、15、17の機能は、コンピュータに実行させることのできるプログラムとし、例えば磁気ディスク（フロッピー（登録商標）ディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体に書き込んでコンピュータに適用可能である。また、このプログラムを通信媒体により伝送してコンピュータに適用することも可能である。上記機能を実現するコンピュータは、このプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

【0126】

【発明の効果】以上詳記したように、本発明においては、スタイルパラメータを入力するのみで容易かつ迅速に出力スタイルを自動生成できる。

【0127】また、コンテンツの出力スタイルを動的に生成できるため、ユーザ毎、あるいは状況に応じた出力スタイルでコンテンツの内容を出力することができ、出力スタイルの自由度を拡大できる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態に係る出力スタイル生成システムの構成を例示するブロック図。

【図2】パラメータ解析部の処理内容を例示する図。

【図3】生成プログラム実行部の処理内容を例示する図。

【図4】コンポーネント取得部の処理内容を例示する図。

【図5】コンポーネント作成部のGUI画面の例を示す図。

【図6】コンポーネントデータベースに登録されているコンポーネントを例示する図。

【図7】同実施の形態に係る出力スタイル生成システムの実行する処理を示すフロー図。

【図8】コンテンツを例示する図。

【図9】本発明の第2の実施の形態においてスタイルパラメータを受け付けた出力スタイル生成システムの具体的な処理内容を示すフロー図。

【図10】スタイルパラメータに対する生成プログラム実行部の具体的な処理内容を例示する図。

【図11】スタイルパラメータに対するパラメータ解析処理部の具体的な処理内容を例示する図。

【図12】コンポーネント取得部の具体的な処理内容を例示する図。

【図13】コンポーネントを例示する図。

【図14】スタイルパラメータ「見出し=下線付き見出し、表示形式=テーブル」に基づいて生成された出力スタイルデータを用いてコンテンツを表示した画面図。

【図15】スタイルパラメータ「見出し=斜体見出し,表示形式=リスト」に基づいて生成され出力スタイルデータを用いてコンテンツを表示した画面図。

【図16】本発明の第3の実施の形態に係る出力スタイル生成システムの構成を例示するブロック図。

【図17】本発明の第4の実施の形態に係る出力スタイル生成システムの構成を例示するブロック図。

【図18】本発明の第5の実施の形態に係る出力スタイル生成システムの利用状況を例示するブロック図。

【図19】計算機システムにおけるコンテンツと出力スタイルデータの取り扱い状態を示すブロック図。

【図20】計算機システムにおけるコンテンツと静的に用意した複数の出力スタイルデータの取り扱い状態を示すブロック図。

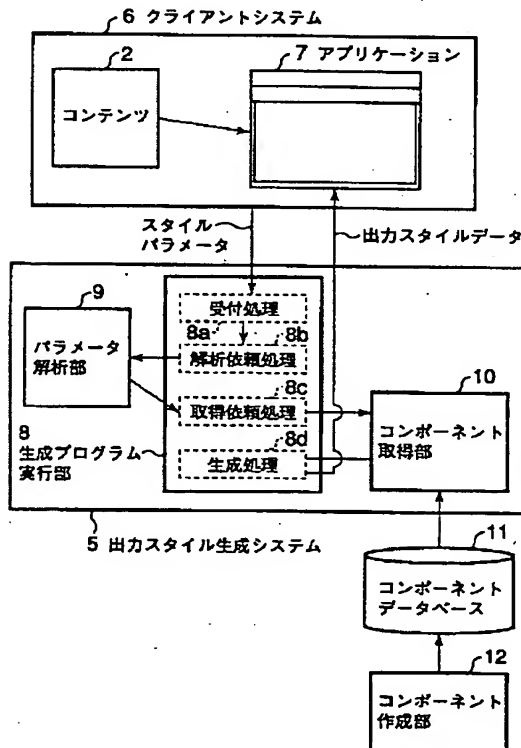
【図21】構造化文書の体裁定義支援装置のシステム構成図。

【符号の説明】

1…ブラウザ

- 2、13…コンテンツ
- 3、3a～3c…出力スタイルデータ
- 4…構造化文書の体裁定義支援装置
- 5、15、17…出力スタイル生成システム
- 6…クライアントシステム
- 7…アプリケーション
- 8、16…生成プログラム実行部
- 8a…受付処理
- 8b…解析依頼処理
- 8c…取得依頼処理
- 8d、16d…生成処理
- 9…パラメータ解析部
- 10…コンポーネント取得部
- 11…コンポーネントデータベース
- 12…コンポーネント作成部
- 18…フィードバック部
- 20…インターネット

【図1】



【図2】

```

9 パラメータ解析部

public static Hashtable analyze(String params){
    //
    paramsは引数であるパラメータを表す。
    ここでは仮に"param1=A,param2=B,param3=C"となっているとする。
    //

    //パラメータをカンマで区切って分離する。
    StringTokenizer st = new StringTokenizer(params,",");

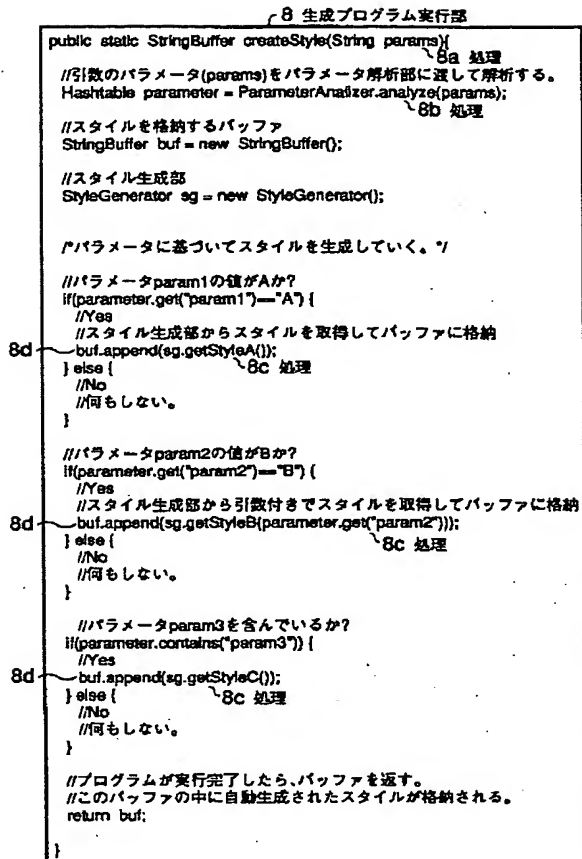
    //パラメータを格納するためのハッシュテーブル。
    Hashtable hash = new Hashtable();

    //パラメータの数だけ繰り返す。
    while(st.hasMoreTokens()){
        //パラメータ("param1=A")を取り出す。
        String param = st.nextToken();
        // "="があるインデックス。
        int index = param.indexOf("=");
        //ハッシュテーブルにキー(param1)とバリュー(A)を格納する。
        hash.put(param.substring(1,index),param.substring(index));
    }

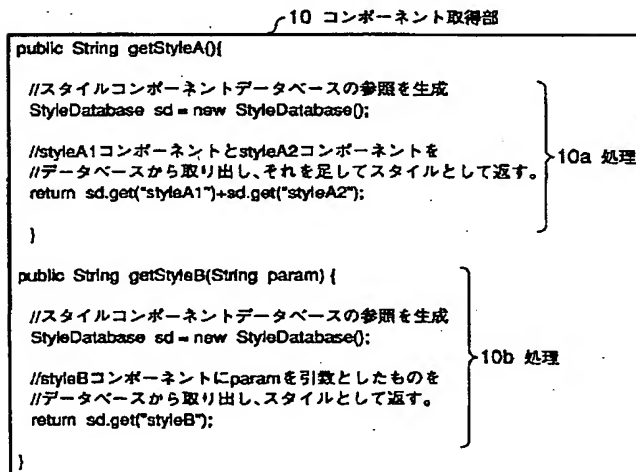
    //ハッシュテーブルを返す。
    ハッシュテーブルには、
    ((param1,A) (param2,B) (param3,C))
    という解析済みパラメータが格納される。
    //
    return hash;
}

```

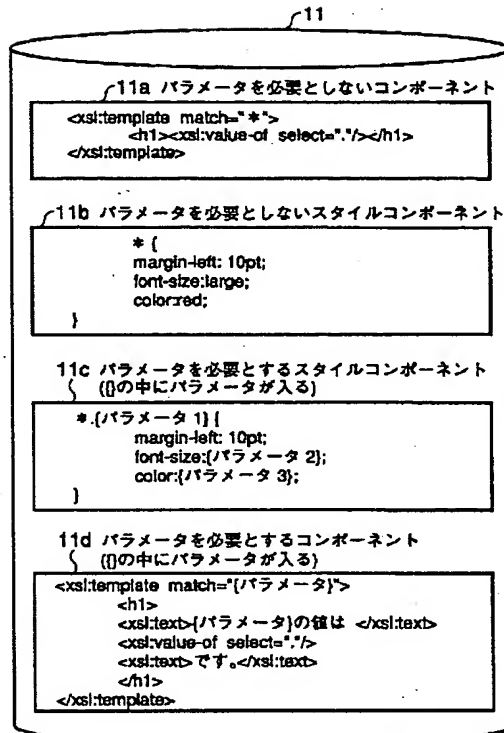
【図 3】



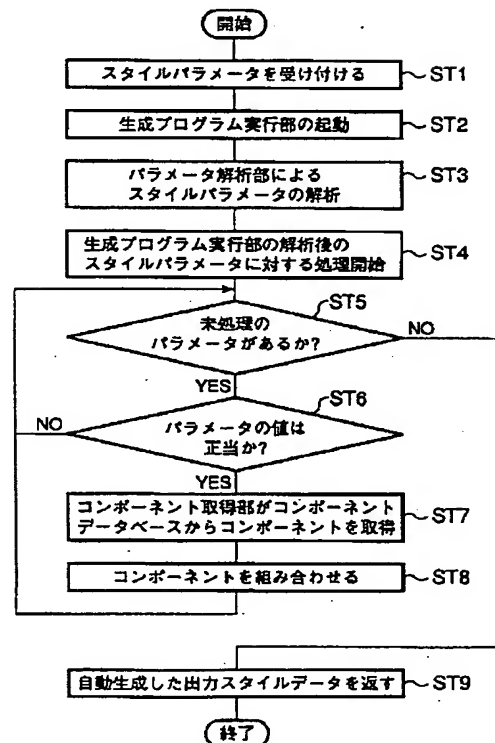
【図 4】



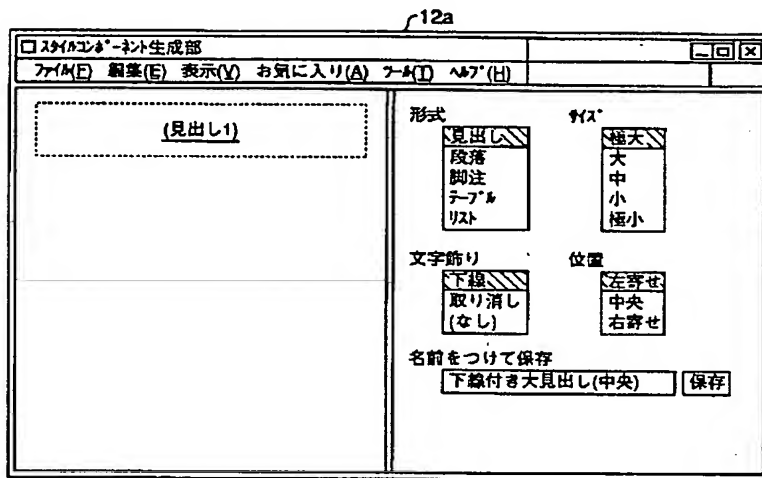
【図 6】



【図 7】



【図5】

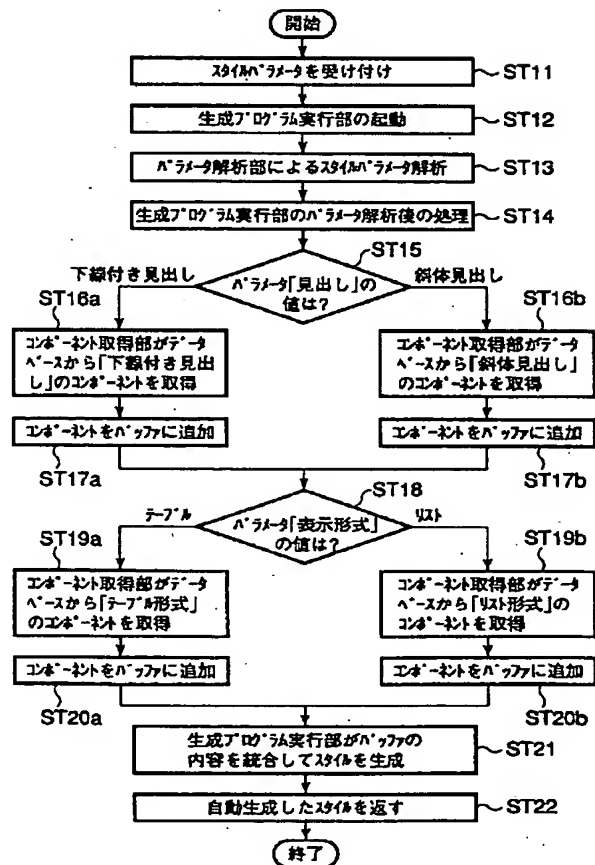


【図8】

13 コンテンツ

```
<?xml version="1.0"?>
<list>
  <name>商品リスト</name>
  <data>
    <item>
      <product>テレビ</product>
      <price>¥50,000</price>
    </item>
    <item>
      <product>ビデオ</product>
      <price>¥40,000</price>
    </item>
    <item>
      <product>DVDプレーヤー</product>
      <price>¥50,000</price>
    </item>
    <item>
      <product>デジタルカメラ</product>
      <price>¥60,000</price>
    </item>
    <item>
      <product>ノートパソコン</product>
      <price>¥250,000</price>
    </item>
    <item>
      <product>洗濯機</product>
      <price>¥30,000</price>
    </item>
    <item>
      <product>掃除機</product>
      <price>¥10,000</price>
    </item>
    <item>
      <product>冷蔵庫</product>
      <price>¥40,000</price>
    </item>
    <item>
      <product>電子レンジ</product>
      <price>¥90,000</price>
    </item>
    <item>
      <product>エアコン</product>
      <price>¥100,000</price>
    </item>
  </data>
</list>
```

【図9】



【図10】

8 生成プログラム実行部

```

public static StringBuffer createStyle(String params){
//引数のパラメータ(params)をパラメータ解析部に渡して解析する。
Hashtable parameter = ParameterAnalyzer.analyze(params);

//スタイルを格納するバッファ
StringBuffer buf = new StringBuffer();

//スタイル生成部
StyleGenerator sg = new StyleGenerator();

//パラメータに基づいてスタイルを生成していく。
//パラメータ「見出し」のチェック
if(parameter.get("見出し")=="下線付き見出し"){
//Yes
//スタイル生成部からスタイルを取得してバッファに格納
buf.append(sg.getStyleUnderlineHeadline());
} else if(parameter.get("見出し")=="斜体見出し"){
//Yes
//スタイル生成部からスタイルを取得してバッファに格納
buf.append(sg.getStyleItalicHeadline());
} else {
//No
//何もしない。
}

//パラメータ「表示形式」のチェック
if(parameter.get("表示形式")=="テーブル"){
//Yes
//スタイル生成部からスタイルを取得してバッファに格納
buf.append(sg.getStyleTable());
} else if(parameter.get("表示形式")=="リスト"){
//Yes
//スタイル生成部からスタイルを取得してバッファに格納
buf.append(sg.getStyleList());
} else {
//No
//何もしない。
}

//プログラムが実行完了したら、バッファを返す。
//このバッファの中に自動生成されたスタイルが格納される。
return buf;
}

```

【図12】

10 コンテキスト取得部

```

public String getStyleUnderlineHeadline(){
//スタイルコンテキストの参照を生成
StyleDatabase sd = new StyleDatabase();

//下線付き見出しのスタイルコンテキストを取得して返す
return sd.get("UnderlineHeadline");
}

public String getStyleItalicHeadline(){
//スタイルコンテキストの参照を生成
StyleDatabase sd = new StyleDatabase();

//斜体見出しのスタイルコンテキストを取得して返す
return sd.get("ItalicHeadline");
}

public String getStyleTable(){
//スタイルコンテキストの参照を生成
StyleDatabase sd = new StyleDatabase();

//テーブル表示のスタイルコンテキストを取得して返す
return sd.get("Table");
}

public String getStyleList(){
//スタイルコンテキストの参照を生成
StyleDatabase sd = new StyleDatabase();

//リスト表示のスタイルコンテキストを取得して返す
return sd.get("List");
}

```

【図11】

9 パラメータ解析部

```

public static Hashtable analyze(String params){
//paramsは引数であるパラメータを表す。
//ここでは「見出し=下線付き見出し、表示形式=テーブル」となっていると看做す。

//パラメータをカンマで区切って分離する。
StringTokenizer st = new StringTokenizer(params, ",");

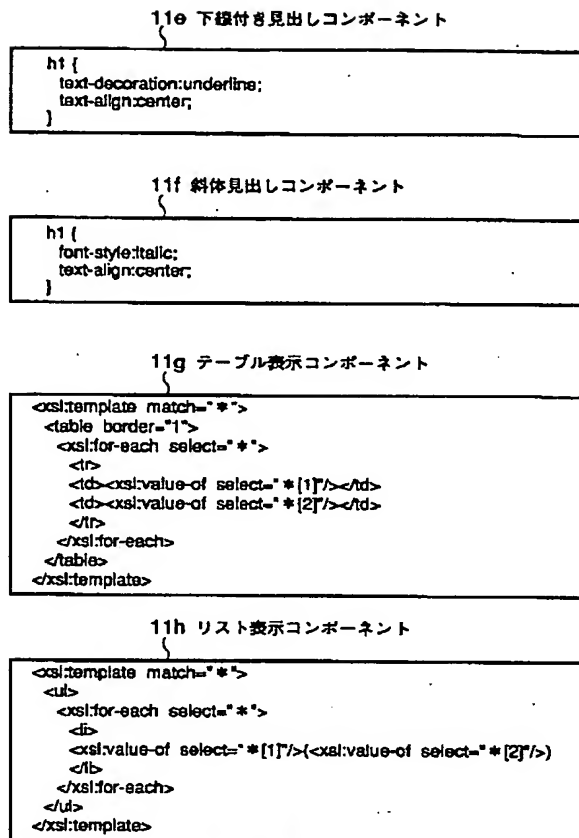
//パラメータを格納するためのハッシュテーブル。
Hashtable hash = new Hashtable();

//パラメータの数だけ繰り返す。
while(st.hasMoreTokens()){
//パラメータ「見出し=下線付き見出し」を取り出す。
String param = st.nextToken();
//"="があるインデックス。
int index = param.indexOf("=");
//ハッシュテーブルにキー(見出し)とバリュー(下線付き見出し)を格納する。
hash.put(param.substring(1, index), param.substring(index));
}

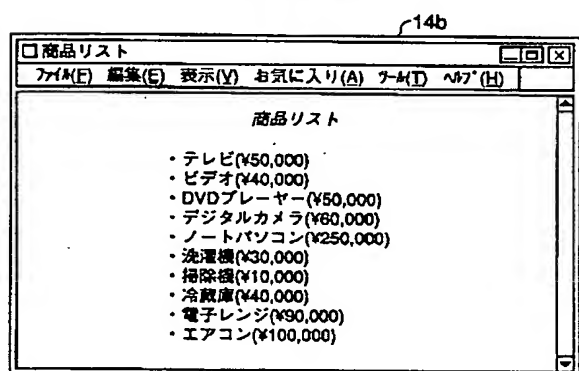
//ハッシュテーブルを返す。
//ハッシュテーブルには、
//{(見出し, 下線付き見出し) (表示形式, テーブル)}
//という解析済みパラメータが格納される。
return hash;
}

```

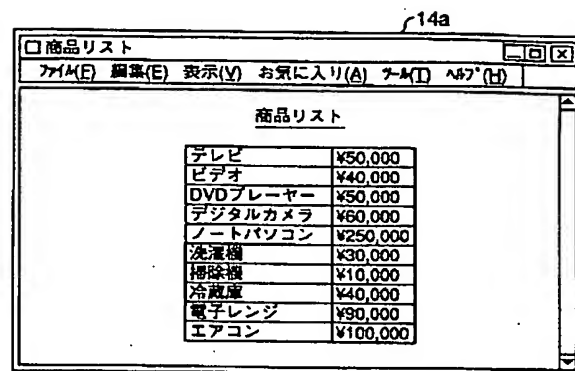
【図13】



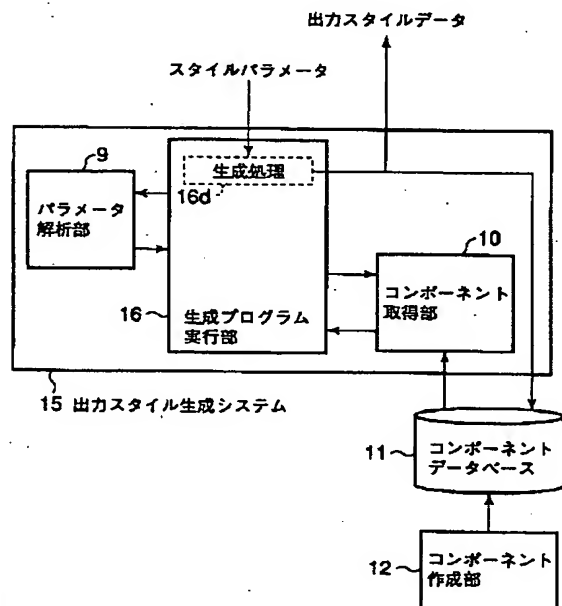
【図15】



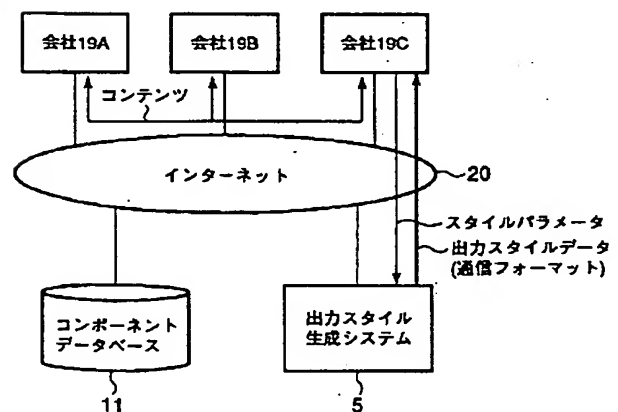
【図14】



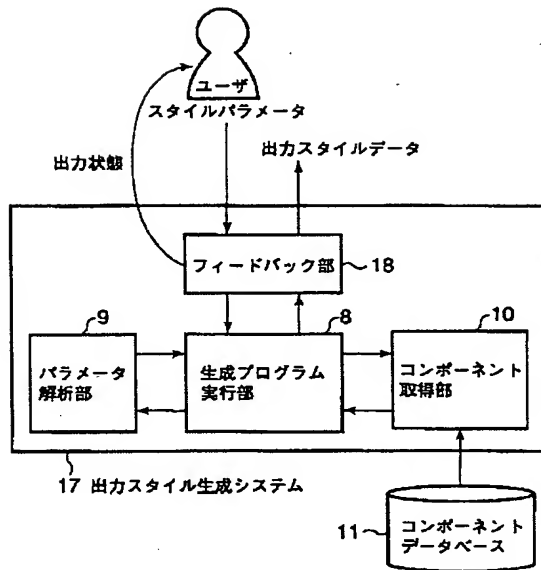
【図16】



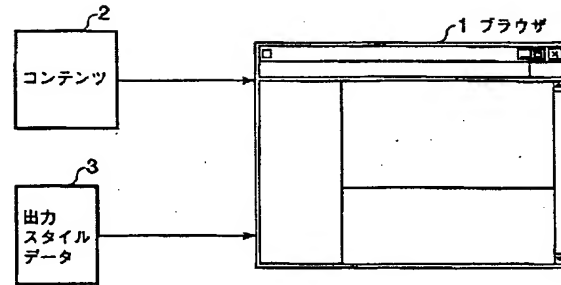
【図18】



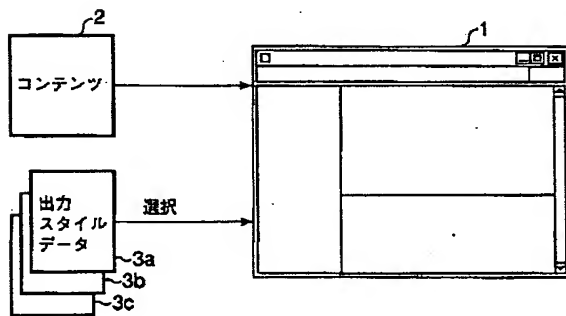
【図17】



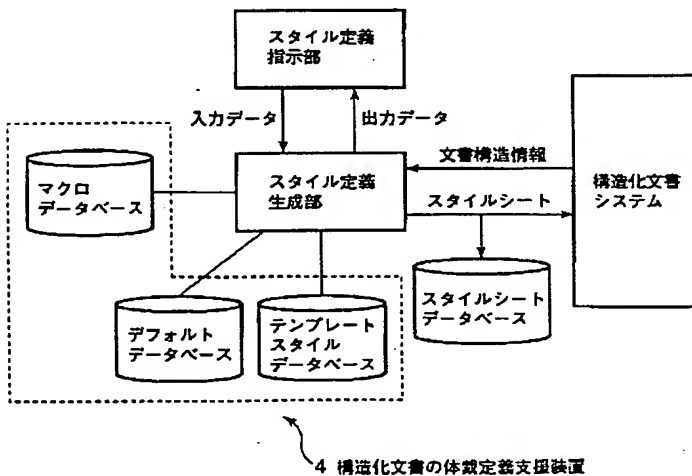
【図19】



【図20】



【図21】



フロントページの続き

(72)発明者 原嶋 秀次

東京都府中市東芝町 1 番地 株式会社東芝

府中事業所内

F ターム(参考) 5B009 NA03 NA05

5B069 BB16 DC00 FA01 JA01

This Page Blank (uspto)